# **Introducción**

El presente documento muestra los resultados de aplicar el Proceso Unificado de Desarrollo al proyecto actual.

# **Índice**

1. Consideraciones iniciales

2. Ciclo de desarrollo

2.1 Fase de inicio

2.2 Fase de elaboración

2.3 Fase de construcción

2.4 Fase de transición

3. Futuros ciclos de desarrollo

4. Casos de uso

5. Diagramas UML

6. Controles de calidad

7. Consideraciones finales

## 1. Consideraciones iniciales

En el proyecto actual se hace uso de un Proceso de Desarrollo Unificado (RUP). Esto obliga a seguir determinadas pautas como la dirección por casos de uso, división de la construcción y versionamiento del producto en una serie de ciclos o el uso de diagramas UML.

## 2. Ciclo de desarrollo

El modelo RUP califica la vida de un sistema como una serie de ciclos. Dadas las limitaciones de tiempo del curso actual, para la asignatura de Ingeniería del Software tan sólo podrá ser llevado a cabo uno de estos ciclos. En él se tratará de llegar a una versión básica y perfectamente funcional del producto.

Este ciclo, siguiendo la teoría del modelo RUP, se divide en varias fases explicadas a continuación. Éstas se encuentran brevemente explicadas en el apéndice del documento de especificación de requisitos software, por lo que en el presente documento nos limitaremos a dar una visión de la aplicación de las mismas al proyecto actual.

### 2.1 Fase de inicio

Esta fase ha ocupado las primeras semanas tras la creación del grupo de trabajo. En ella se han llevado a cabo reuniones más frecuentes para hacer brainstorming de ideas y delimitar lo máximo posible la estructura básica del producto (Jugador Numero 12).

Se ha dedicado bastante tiempo a esta fase ya que se consideraba importante para que todos los miembros del grupo tuviesen suficientemente claro el enfoque y funcionalidades del producto para poder trabajar cómodamente.

### 2.2 Fase de elaboración

Al igual que en el caso anterior, esta fase ha ocupado otras pocas semanas del proyecto. Se han llevado a cabo reuniones regulares con diversos fines:

* Selección de las tecnologías y frameworks a emplear. Hubo bastantes debates con respecto al framework de PHP a emplear. Se presentaron dos alternativas (Yii y Symfony/Silex). Finalmente se decidió hacer uso de Yii.
* Selección del repositorio y alojamiento de la documentación. En este aspecto se decidió de forma unánime el uso de GitHub en lo referente al código de la aplicación y Google Drive (con Google Docs) para la documentación.
* Establecer unas bases arquitectónicas sólidas para la creación del producto.
* Instrucción a los miembros del grupo en las tecnologías a usar mediante clases privadas en reuniones, emails de ayuda y tutoriales en la documentación.
* Creación de nuevos roles. Al margen del papel de líder de grupo, que, como es obvio, ya había sido establecido, se crearon nuevos roles:  
  + El denominado “bibliotecario”, que se encargaría de la supervisión de la estructura y coherencia de la documentación. Este rol iría cambiando cada semana para evitar que un único miembro del grupo cargase con esta tediosa tarea durante demasiado tiempo.
  + El supervisor de código, cuya función principal sería garantizar la corrección del código creado para la aplicación y asesorar en caso de dudas de implementación. Esto es, que se respete el patrón MVC, que se eliminen las máximas brechas de seguridad posibles, que se garantice la estabilidad del sistema, etc.
  + El supervisor de repositorio, que se ocuparía de garantizar la coherencia en los commits del repositorio. Por ejemplo, que no se suban fragmentos incoherentes, que se respete la organización de ramas, etc.

A medida que el proyecto avanzaba y todo se empezaba a hacer de forma correcta (los commits organizados, ramas respetadas, código decente dentro de unos límites, etc) este sistema de roles se ha ido relegando a un segundo plano.

* Asignación de tareas. A cada miembro del grupo le fueron asignadas una serie de tareas iniciales para comenzar la construcción del producto.

### 2.3 Fase de construcción

Siguiendo una estructura de iteraciones sobre el producto, cerca del mes de diciembre se empezó con la implementación del sistema. Para ello se partió de una base muy simple que incluía el sistema de sesiones y el esqueleto básico de la aplicación. Sobre ella se irían agregando diversos módulos directamente relacionados con las funcionalidades/casos de uso y que, una vez unidos, conformarían el sistema terminado.

Para cada iteración se ha establecido una breve fase de análisis de requisitos y diseño, posteriormente se ha procedido a la implementación y finalmente a las pruebas de la misma.

* En una primera iteración se fueron creando elementos básicos como el sistema de registro, la visualización de datos básicos y la modificación de los mismos (todo dejando de lado el apartado de presentación). Algunos miembros del grupo fueron asignados a estas tareas de implementación, otros a la elaboración de la documentación y una minoría continuó planificando los futuros módulos a implementar. Durante esta iteración se ralentizó un poco el desarrollo del sistema pero se consiguió documentación que serviría de consulta para posteriores dudas.
* Una vez finalizada esta primera fase, en una segunda iteración, se procedió a agregar el módulo del partido, las acciones individuales/grupales/pasivas, algunas funcionalidades como el cambio de equipo y el nuevo diseño del estilo de la página. Esta iteración duró algo más que la primera dada la complejidad de ciertas implementaciones y constituyó los elementos presentes en la demo del mes de marzo.
* Posteriormente, en una tercera iteración, que comenzó en el mes de abril, se incorporaron nuevas funcionalidades. Entre las principales podemos destacar la interacción durante el partido (acciones de partido), la recarga automática de éste (incorporación de Ajax a varios niveles), el sistema de mensajería privada y el de notificaciones. Asimismo se agregaron otros elementos como el chat durante el encuentro o las estadísticas del mismo. Por último, se generó un nuevo estilo para toda la web, mucho más atractivo que el anterior.  
  Todos estos contenidos se organizaron para estar presentes en la demo del día 28 de abril.
* En último lugar encontramos la cuarta iteración de la fase de construcción. Ésta se dedicó exclusivamente a reparación completa de errores, retoques finales del estilo y finalización de la documentación. Se incorporaron pequeños módulos como el tutorial o el cooldown de acciones de partido.

El resultado de todos estos cambios se presentó en la demo del día 22 de mayo.

### 2.4 Fase de transición

Durante esta última fase se llevaron a cabo todos los preparativos para la entrega final: desde reparar pequeños bugs encontrados en el producto hasta finalizar/actualizar los últimos documentos necesarios.

Asimismo, se generó el material necesario para el acto oficial del día 4 de junio y se presentó el proyecto en el mismo.

## 3. Futuros ciclos de desarrollo

Una vez finalizado este primer ciclo y el curso de Ingeniería del Software, el sistema será, casi con toda seguridad, llevado a un segundo ciclo de desarrollo al margen de la asignatura. En él se retomarán de nuevo con las 4 fases anteriores y se refinará y mejorará el producto inicial para garantizar una implementación suficientemente decente como para salir al mercado sin problemas.

## 4. Casos de uso

Se han creado varios documentos con los casos de uso asociados a la funcionalidad del sistema. En una primera versión presentaban los fragmentos de funcionalidad de forma muy básica pero posteriormente, en base a ejemplos reales de casos de uso, fueron refinados y detallados con más precisión.

Se optó por tener muchos casos de uso simples con el fin de mejorar su comprensión. Esto ha permitido poder unir varios de ellos para crear diferentes módulos de funcionalidad a implementar en el sistema. Otra opción hubiese sido crear pocos casos de uso pero con mucho más contenido y funcionalidad, pero era algo que resultaba bastante engorroso de consultar.

Podemos notar, además, que los casos de uso están muy relacionados con los diagramas de flujo de ejecución. Esto permite que, en caso de duda, se pueda consultar el diagrama (ya que es más directo) y, si hace falta información más detallada, se pase al caso de uso correspondiente.

Este tipo de documentación ha servido para dirigir el diseño de las iteraciones del producto, ya que se tomaban casos de uso restantes, se agrupaban en funcionalidades, y se acoplaban a las nuevas iteraciones de la fase de construcción.

## 5. Diagramas UML

La generación de estos diagramas comenzó bastante tarde. No se llevaron a cabo durante las etapas oportunas del desarrollo. Más bien, se podría decir que se generaron como documentación adicional con el principal fin de aprender a crearlos. Por supuesto, también estaban a disposición de todo aquel que quisiera consultarlos, pero, francamente, no se han usado para mucho más que con fines de aprendizaje.

No obstante, de esto sí hemos aprendido que, realizados a tiempo, pueden ser de gran utilidad (al igual que toda la documentación más “visual” como diagramas de flujo). Con un simple vistazo permiten diferenciar claramente estructuras del sistema.

## 6. Controles de calidad

Dado que la fase de construcción comenzó de forma algo retrasada, no se llevaron a cabo reuniones específicas de control de calidad del código y documentación. En lugar de esto, dichas supervisiones fueron llevadas a cabo atendiendo a los siguientes puntos:

* Horas de laboratorio: se aprovechaban para juntar y revisar todo el código funcional hecho durante la semana. Se optó por este método ya que era la forma ideal de reunir a todo el grupo sin tener que fijar horario extraacadémicos.
* Reuniones de implementación: usadas principalmente para resolver posibles dudas de implementación, continuar con la construcción de funcionalidades y revisar el código creado.
* Revisión por parte de miembros del grupo: al margen de las horas de laboratorio y reuniones de implementación, dos componentes del equipo (Javier López y Manuel Artero) se han encargado de ir revisando continuamente las implementaciones realizadas. Para este apartado ha resultado especialmente útil el rol de “supervisor de código” descrito en un punto anterior.
* Documentación: no se han realizado reuniones estrictas para este tema. El principal control de calidad ha sido el aportado por el rol de “bibliotecario” (descrito también en un punto anterior). A pesar de existir un periodo más intenso de implementación en el cuál la documentación ha sido relegada a un segundo plano, se ha hecho un esfuerzo por retomar toda la documentación y actualizarla en la medida de lo posible.

## 7. Consideraciones adicionales

Esta sección incluye una breve opinión y conclusiones obtenidas de la aplicación del Proceso Unificado de Desarrollo al proyecto actual. Se dividirá en varios puntos para mejorar su comprensión:

* En primer lugar, mencionar que, al principio, tanto la fase de inicio como la de elaboración (en menor medida) parecían una pérdida de tiempo. Lo mismo ocurría con la documentación. El tiempo pasaba y los proyectos que seguían metodologías ágiles avanzaban rápidamente mientras que el nuestro permaneció “estancado” hasta principios de diciembre. Ahora bien, una vez el proyecto avanzaba se iba notando la importancia de estas fases iniciales. Dado que en los primeros meses se habían dejado muy claras las bases del producto, la fase de construcción pudo centrarse tan sólo en la implementación, sin necesidad de reuniones de última hora o preguntas acerca de cómo funciona el sistema (salvo algunas puntuales que siempre surgen). Asimismo, la documentación (en especial los diagramas UML) servían como buena referencia a la hora de diseñar e implementar los diferentes módulos.
* Al principio, el modelo RUP resultaba algo caótico, hasta el punto de no saber si se estaba aplicando correctamente. Una vez que el proyecto avanzaba y se repasaban las bases de dicho proceso, quedaba cada vez más claro cómo funcionaba este modelo y se veían errores cometidos al aplicarlo por primera vez.
* Es muy posible que, de poder volver atrás en el tiempo, hubiésemos seleccionado una metodología ágil como proceso de desarrollo. Ésta nos hubiese permitido llegar a la presentación final con más funcionalidades sin tener que recortar tantas.   
  Otra posible solución, en caso de futuros ciclos del proyecto, sería reducir el tiempo dedicado a las fases de inicio/elaboración o bien hacerlas más productivas.   
  No obstante, este ciclo RUP ha servido, sin duda, para poder gestionar de manera correcta nuevos ciclos o procesos de desarrollo.
* Han quedado bastante claras las diferencias entre algunos procesos de desarrollo. Por ejemplo, en el caso de trabajar para una “startup” que necesite un producto cuanto antes, es muy posible que lo recomendable sea aplicar una metodología ágil para terminar el sistema antes de la fecha límite y después, una vez lanzado, se puede refinar toda la documentación existente con más calma.  
  En el caso de proyectos de mayor envergadura o grupos de trabajo grandes (o en los que los programadores son desconocidos) es mejor aplicar el modelo RUP ya que permite generar unas bases muy sólidas para el proyecto y basarse en ellas para ir realizando pequeñas iteraciones de funcionalidades.
* Otro detalle importante aprendido es a no dejar de lado la documentación. A lo largo del proyecto las ideas iniciales van cambiando y con ellas las decisiones de diseño. Es muy complicado sentar las bases de un proyecto que permanezca inmutable a lo largo de su desarrollo. Por estos cambios, es conveniente revisar periódicamente la documentación existente y actualizarla.   
  Uno de nuestros errores ha sido dejar dicha documentación de lado una vez iniciada la fase de construcción. El problema llegó a la hora de enfrentarse a una documentación sin actualizar, algo que podía haberse evitado con pequeñas revisiones periódicas de la misma.